USING VIRTUAL IDENTIFIERS TO PROCESS RECEIVED DATA ROUTED THROUGH A NETWORK

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims the benefit of U.S. Provisional Application No. 60/287,069 entitled "METHOD FOR IMPLEMENTING A CLUSTER NETWORK FOR HIGH PERFORMANCE AND HIGH AVAILABILITY USING A FIBRE CHANNEL SWITCH FABRIC," filed April 27, 2001; U.S. Provisional Application No. 60/287,120 entitled "MULTI-PROTOCOL NETWORK FOR ENTERPRISE DATA CENTERS," filed April 27, 2001; U.S. Provisional Application No. 60/286,918 entitled "UNIFIED ENTERPRISE NETWORK SWITCH (UNEX) PRODUCT SPECIFICATION," filed April 27, 2001; U.S. Provisional Application No. 60/286,922 entitled "QUALITY OF SERVICE EXAMPLE," filed April 27, 2001; U.S. Provisional Application No. 60/287,081 entitled "COMMUNICATIONS MODEL," filed April 27, 2001; U.S. Provisional Application No. 60/287,075 entitled "UNIFORM ENTERPRISE NETWORK SYSTEM," filed April 27, 2001; U.S. Provisional Application No. 60/314,088 entitled "INTERCONNECT FABRIC MODULE," filed August 21, 2001; U.S. Provisional Application No. 60/314,287 entitled "INTEGRATED ANALYSIS OF INCOMING DATA TRANSMISSIONS," filed August 22, 2001; U.S. Provisional Application No. 60/314,158 entitled "USING VIRTUAL IDENTIFIERS TO ROUTE TRANSMITTED DATA THROUGH A NETWORK," filed August 21, 2001, and is related to U.S. Patent Application No ___________ entitled "METHOD AND SYSTEM FOR VIRTUAL ADDRESSING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048019US1); U.S. Patent Application No. _________ entitled "METHOD AND SYSTEM FOR LABEL TABLE CACHING IN A ROUTING DEVICE," (Attorney Docket No. 030048024US); U.S. Patent Application No. ___________ entitled "METHOD AND SYSTEM FOR MULTIFRAME BUFFERING IN A ROUTING DEVICE,"

(Attorney Docket No. 030048025US); U.S. Patent Application No. ______________ entitled "METHOD AND SYSTEM FOR DOMAIN ADDRESSING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048026US); U.S. Patent Application No. __________ entitled "METHOD AND SYSTEM FOR INTERSWITCH LOAD BALANCING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048027US); U.S. Patent Application No. ____________ entitled "METHOD AND SYSTEM FOR INTERSWITCH DEADLOCK AVOIDANCE IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048028US); U.S. Patent Application No. _________ entitled "METHOD AND SYSTEM FOR CONNECTION PREEMPTION IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048029US); U.S. Patent Application No. _________ entitled "METHOD AND SYSTEM FOR MULTICASTING IN A ROUTING DEVICE," (Attorney Docket No. 030048030US); U.S. Patent Application No. ___________ entitled "METHOD AND SYSTEM FOR NETWORK CONFIGURATION DISCOVERY IN A NETWORK MANAGER," (Attorney Docket No. 030048032US); U.S. Patent Application No. ______________ entitled "METHOD AND SYSTEM FOR PATH BUILDING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048033US); U.S. Patent Application No. _____________ entitled "METHOD AND SYSTEM FOR RESERVED ADDRESSING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048035US); U.S. Patent Application No. _________ entitled "METHOD AND SYSTEM FOR RECONFIGURING A PATH IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048036US1); U.S. Patent Application No. __________ entitled "METHOD AND SYSTEM FOR ADMINISTRATIVE PORTS IN A ROUTING DEVICE," (Attorney Docket No. 030048037US); U.S. Patent Application No. _________ entitled "PARALLEL ANALYSIS OF INCOMING DATA TRANSMISSIONS," (Attorney Docket No. 030048038US); U.S. Patent Application No. ___________ entitled "INTEGRATED ANALYSIS OF INCOMING DATA TRANSMISSIONS," (Attorney Docket No. 030048039US); U.S. Patent Application No. __________ entitled "USING

VIRTUAL IDENTIFIERS TO ROUTE TRANSMITTED DATA THROUGH A NETWORK," (Attorney Docket No. 030048040US); U.S. Patent Application No. __________ entitled "USING VIRTUAL IDENTIFIERS TO PROCESS RECEIVED DATA ROUTED THROUGH A NETWORK," (Attorney Docket No. 030048041US); U.S. Patent Application No. ____________ entitled "METHOD AND SYSTEM FOR PERFORMING SECURITY VIA VIRTUAL ADDRESSING IN A COMMUNICATIONS NETWORK," (Attorney Docket No. 030048042US); and U.S. Patent Application No. _________ entitled "METHOD AND SYSTEM FOR PERFORMING SECURITY VIA DE-REGISTRATION IN A COMMUNICATIONS NETWORK" (Attorney Docket No. 030048043US), which are all hereby incorporated by reference in their entirety.

TECHNICAL FIELD

[0002] The following disclosure relates generally to computer networks, and more particularly to using virtual identifiers to route data through networks.

BACKGROUND

[0003] The Internet has emerged as a critical commerce and communications platform for businesses and consumers worldwide. The dramatic growth in the number of Internet users, coupled with the increased availability of powerful new tools and equipment that enable the development, processing, and distribution of data across the Internet, have led to a proliferation of Internet-based applications. These applications include e-commerce, e-mail, electronic file transfers, and online interactive applications. As the number of users of and uses for the Internet increases, so does the complexity and volume of Internet traffic. Because of this traffic and its business potential, a growing number of companies are building businesses around the Internet and developing mission-critical business applications to be provided by the Internet.

[0004] Existing enterprise data networks ("EDNs") that support e-commerce applications are straining under the demand to provide added performance and

services to customers. In particular, the growing customer demands for services have resulted in increasingly complex ad hoc EDNs. Current architectures of EDNs typically include three sub-networks: 1) a web server local area network (LAN), 2) a computational network for application servers, and 3) a storage area network (SAN). The processing and storage elements attached to these sub-networks may have access to a wide area network (WAN) or metropolitan area network (MAN) through a bridging device commonly known as an edge switch. Unfortunately, each of these sub-networks typically uses a distinct protocol and associated set of hardware and software, including network interface adapters, network switches, network operating systems, and management applications. Communication through the EDN requires bridging between the sub-networks that requires active participation of server processing resources for protocol translation and interpretation. There are a variety of disadvantages to the current architecture of EDNs, many of which result because the sub-networks and associated applications are developed by different vendors and it is difficult to integrate, manage, maintain and scale such inter-vendor EDNs. The ability to provide affordable, high-performance EDN solutions with extensive scalability, very high availability, and ease of management is thus significantly compromised or completely lost as existing solutions are grown ad hoc to meet customer demands.

[0005] In addition to inter-vendor problems that exist in current EDN architectures, it is often difficult to transmit data to appropriate destinations in a secure manner, particularly with any guarantees as to the Quality Of Service ("QOS") of the transmissions. For example, current architectures typically assign one or more network addresses to each node in a network (*e.g.*, logical network addresses such as IP addresses and/or physical network addresses such as Media Access Control ("MAC") addresses), and network routing and switching devices use the network addresses of a destination node to route transmissions of data from a source node to that destination node. However, it is difficult to prevent unauthorized source nodes from sending data to a destination node with a known

network address, particularly if the source nodes masquerade their identities by spoofing their own network addresses, and correspondingly it is difficult for a destination node to ensure that received data is from an authorized source. In addition, it can be difficult for even an authorized source node to transmit data to desired destinations, as the source node must know the appropriate network address or other logical name (*e.g.*, a DNS name) that is assigned or mapped to a destination node in order to perform the transmitting. Even more difficult are situations in which the appropriate destinations are difficult to identify, such as for a source node that is publishing data of a type that may be of interest to various potential subscriber destination nodes. Finally, current architectures typically do not allow a source node to ensure that transmitted data will be processed with a desired QOS, such as a minimum network latency or minimum level of throughput.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Figure 1 is a network diagram illustrating various nodes of an example Fibre Channel Fabric network that are inter-communicating.

[0007] Figures 2A-2C illustrate an example of Virtual Identifier Network Interface Controller ("NIC") embodiments using virtual identifiers to inter-communicate through an example Fibre Channel Fabric network.

[0008] Figure 3 is a block diagram illustrating a node using an embodiment of the disclosed Virtual Identifier NIC to communicate with other nodes.

[0009] Figure 4 is a flow diagram of an embodiment of the Communication Registrar routine.

[0010] Figure 5 is a flow diagram of an embodiment of the Outgoing Communication Translator routine.

[0011] Figure 6 is a flow diagram of an embodiment of the Verify Communication Transmittal subroutine.

[0012] Figure 7 is a flow diagram of an embodiment of the Incoming Communication Translator routine.

DETAILED DESCRIPTION

[0013] A software facility is described below that uses virtual identifiers to route communications through a network to destinations in an appropriate manner. In some embodiments, each virtual identifier is assigned to a path through a network to a destination, such as by a network manager for the network. Using virtual identifiers for routing of communications, rather than network addresses or logical names that are specific to a destination, provides a variety of benefits, as discussed in greater detail below.

[0014] In some embodiments, one or more Virtual Identifier ("VI") Network Interface Controller ("NIC") facilities on each node (*e.g.*, one VI NIC for each network interface) facilitate the use of virtual identifiers in communicating data. When a VI NIC on a node receives an indication that a data communication to one or more remote nodes is to occur, such as from an application executing on the node, the VI NIC will identify an appropriate transmittal virtual identifier that can be used to route the data communication through the network to the appropriate remote destination nodes without being assigned to or directly associated with those destination nodes. Such data communications can include both transitory connectionless transmittals of data (*e.g.*, unidirectional transmittals from a source to a destination) and non-transitory connections that allow multiple distinct transmittals of data (*e.g.*, a persistent dedicated connection that allows a connection-initiating source and a connection destination to transmit data back and forth).

[0015] The VI NIC can identify an appropriate transmittal virtual identifier for routing a data communication in various ways. In some embodiments, the VI NIC will register some or all outgoing data communications with a network manager for the network, and will receive an appropriate transmittal virtual identifier to be used for that communication from the network manager. If an indicated data communication corresponds to a previously registered data communication (*e.g.*, to an existing connection or to a previous communication to the same destination

and in the same transmission manner), however, the VI NIC could instead in some embodiments use the previously received transmittal virtual identifier for that data communication rather than perform an additional registration for the indicated data communication. The manners in which a data communication can be transmitted vary with the transmission characteristics that are supported by a network, and can include factors such as a particular Class Of Service ("COS") or transmission priority.

[0016] In some embodiments in which virtual identifiers are assigned to paths through a network, the assignment of paths to such virtual path identifiers is performed in a dynamic fashion after an indication is received that a data communication is to occur, such as by the network manager upon receipt of a data communication registration. The assigning of a virtual path identifier to a path can include the configuring of each of one or more intermediate routing devices (*e.g.*, routers or switches) between the source and the destination, such as by the network manager, so that when one of the routing devices receives a data communication that includes the virtual identifier it will forward the communication in an appropriate manner either directly to the destination or instead to a next routing device along the path that is similarly configured.

[0017] The VI NIC can also assist in determining appropriate destinations for an indicated data communication, either directly or in conjunction with the network manager (*e.g.*, by registering the data communication with the network manager), with the transmittal virtual identifier for that data communication selected so as to route the data communication to those destinations. In some situations, the indicated data communication may explicitly specify a destination, such as with a destination network address, while in other situations a destination may not be specified, such as when an application is publishing information and is relying on a third party to route the information to one or more current subscribers for that information. Regardless of whether a destination is specified, however, the VI NIC and/or the network manager can select one or more destinations that are appropriate for the indicated data communication, even if the specified destination

is not among the selected destinations. This destination selection can be made by considering one or more of various factors, including any destinations specified, any expressions of interest made by potential recipients in the data communication (*e.g.*, subscription requests), the type of data being communicated, the manner of the data communication (*e.g.*, a specified COS and/or transmission priority), the identity or type of the source node and/or source application, the type of a destination application, etc.

[0018] In some situations, a source of an indicated data communication may specify a destination using a destination network address that is not mapped to any node in the network, and if so the VI NIC and/or the network manager could then select an appropriate destination for that destination network address. Multiple destinations can also be selected for an indicated data communication, even if that data communication specified a single destination (which may or may not be one of the selected destinations). If so, a single transmittal virtual identifier can be used to route the data communication to each of the multiple selected destinations, such as by configuring one or more intermediary routing devices to divide received communications that use that transmittal virtual identifier so as to forward a copy of such received communications to each of multiple destinations (or multiple next routing devices).

[0019] In some embodiments, virtual identifiers correspond to paths through a network that are specific to a source. If so, a single virtual identifier can be used by different sources for different paths, such as to different destinations if the different paths do not overlap. The use of virtual addresses also allows a path corresponding to a virtual identifier to be reconfigured in a manner transparent to a source using that virtual identifier, such as to correspond to a different path to the same destination or to a path to a different destination.

[0020] In some embodiments, when a data communication indicated by a source can result in bi-directional communication (*e.g.*, a response from one or more of the destinations), the VI NIC also identifies a response virtual identifier that can be used for routing data from one or more of the destinations back to the source.

If the VI NIC registers the data communication with a network manager, this response virtual identifier may be received from the network manager. After identifying this response virtual identifier, the VI NIC associates it with information indicating how to process received data communications that are routed using the response virtual identifier. In some embodiments, such received data communications are processed by forwarding the data communications to one or more resources associated with the destination node, such as an executing application program, a file on storage, or a device that is part of the node. For example, if a source application on a source node initiates a bi-directional communication, a VI NIC for the source node may associate the response virtual identifier with that source application so that received responses can be forwarded to that source application (which then becomes the destination application for those received communications).

[0021] The association of a virtual identifier with a corresponding destination application to which a data communication will be forwarded can be performed in various ways. For example, software applications that communicate using TCP/IP mechanisms often use TCP/IP sockets, which include a combination of an IP address and a software port number specific to a computing device using that IP address. Thus, in those embodiments the response virtual identifier can be associated with socket information for the source application. In a similar manner, in some embodiments a destination node associates transmittal virtual identifiers used to route data communications to that destination with an appropriate resource local to the destination node, such as based on information provided to the destination node by the network manager as part of the registering of those data communications and/or based on information included as part of the data communications.

[0022] When the VI NIC has access to application-specific information for a destination application for a received communication, such as TCP/IP socket information that is associated with a response virtual identifier, the VI NIC can use the information to provide additional benefits. For example, many network nodes

and/or applications executing on such nodes require that various information be correctly specified in a received communication in order for that communication to be accepted, such as for security reasons. One example is that a destination application using TCP/IP communication mechanisms may require that any received transmissions include the correct TCP/IP socket information corresponding to that application. However, the previously discussed use of transmittal virtual identifiers can result in valid communications being received having incorrect TCP/IP socket information for a destination application, as discussed in greater detail below. When this occurs, the VI NIC that receives the communication can replace the incorrect included TCP/IP socket information with the correct information for the application by using the TCP/IP socket information that is associated with the transmittal virtual identifier used to route the communication. In addition, in some embodiments the VI NIC may verify the accuracy of the received communication in various ways before performing such information replacement.

[0023] The use of virtual identifiers can result in valid received communications that have incorrect information for a destination application in various ways. For example, if a source application specifies a destination IP address and that destination IP address is included in the data being communicated (*e.g.*, in a location reserved for such a destination network address), but a VI NIC for that source application identifies one or more destinations that do not correspond to that destination IP address (*e.g.*, that have other IP addresses), then the data communication will include a specified destination IP address that does not correspond to the IP addresses used by applications at the identified destinations. In addition, if multiple destinations with different IP addresses are identified by the VI NIC when only a single destination IP address was specified, most of the destinations will receive communications that do not include correct IP address information. In such situations, the VI NIC that receives the communication can replace the incorrect included IP address information with the correct IP address information for the application by using the TCP/IP socket information that is

associated with the virtual identifier used to route the communication. Those skilled in the art will appreciate that a similar information replacement can be used for other communication mechanisms. In addition, in situations in which a data communication is being routed to only a single destination, the VI NIC that sends the data communication can perform the information replacement if that VI NIC has access to the necessary application-specific information for the destination application.

[0024] In some embodiments, a VI NIC can also identify information related to routing a data communication other than a transmittal virtual identifier, either directly or in conjunction with the network manager (*e.g.*, by registering the data communication with the network manager). For example, the VI NIC may identify one or more Quality Of Service ("QOS") parameters that relate to a manner in which the data communication should occur, such as a specified COS and/or a priority to be used for the transmission of the data. If so, the VI NIC can also use such QOS parameters when transmitting data for that data communication.

[0025] Additional details about virtual identifiers and their uses by network managers and network routing devices are discussed in the following patent applications, each of which are incorporated by reference in their entirety: Provisional U.S. Application No. 60/287,068, filed April 27, 2001, entitled "GENERATION OF SYNCHRONIZED 50% DUTY CYCLE CLOCKS" (attorney docket no. 030048011US); Provisional U.S. Application No. 60/287,121, filed April 27, 2001, entitled "FREQUENCY DETECTION AND LOCK FOR PHASED LOCK LOOP" (attorney docket no. 030048012US); Provisional U.S. Application No. 60/287,069, filed April 27, 2001, entitled "METHOD FOR IMPLEMENTING A CLUSTER NETWORK FOR HIGH PERFORMANCE AND HIGH AVAILABILITY USING A FIBRE CHANNEL SWITCH FABRIC" (attorney docket no. 030048013US); Provisional U.S. Application No. 60/287,120, filed April 27, 2001, entitled "MULTI-PROTOCOL NETWORK FOR ENTERPRISE DATA CENTERS" (attorney docket no. 030048014US); Provisional U.S. Application No. 60/286,918, filed April 27, 2001, entitled "UNIFIED ENTERPRISE NETWORK SWITCH

(UNEX) PRODUCT SPECIFICATION" (attorney docket no. 030048015US); Provisional U.S. Application No. 60/286,922, filed April 27, 2001, entitled "QUALITY OF SERVICE EXAMPLE" (attorney docket no. 030048016US); Provisional U.S. Application No. 60/287,081, filed April 27, 2001, entitled "COMMUNICATIONS MODEL" (attorney docket no. 030048017US); and Provisional U.S. Application No. 60/287,075, filed April 27, 2001, entitled "UNIFORM ENTERPRISE NETWORK SYSTEM" (attorney docket no. 030048018US). Each of the following patent applications similarly include additional details about integrating multiple data communication processing techniques and about the use of virtual identifiers, and are also each hereby incorporated by reference in their entirety: Provisional U.S. Application No. 60/314,088 (attorney docket no. 030048015US1), filed August 21, 2001 and entitled "INTERCONNECT FABRIC MODULE"; and Provisional U.S. Application No. 60/314,287, filed August 22, 2001 and entitled "INTEGRATED ANALYSIS OF INCOMING DATA TRANSMISSIONS".

[0026] For illustrative purposes, some embodiments are described below in which the VI NIC is used as part of a Fibre Channel network and/or as part of an EDN architecture. However, those skilled in the art will appreciate that the techniques of the invention can be used in a wide variety of other situations and with other types of networks, including InfiniBand-based networks, and that the invention is not limited to use in Fibre Channel networks or with EDN architectures. Additional details about Fibre Channel are available in “Fibre Channel: A Comprehensive Introduction,” which is authored by Robert W. Kembel and published by Northwest Learning Associates, Inc., and which is hereby incorporated by reference in its entirety. Additional details about InfiniBand is available in the “InfiniBand Architecture Specification, Volumes 1 and 2, Release 1.0.a“, dated June 19, 2001 and available at the time of this writing at the website for the InfiniBand Trade Association at “www.infinibandta.org”, and which is hereby incorporated by reference in its entirety.

[0027] Figure 1 is a network diagram illustrating various nodes of an example Fibre Channel fabric-based interconnect network that are inter-communicating using virtual identifiers. In this example embodiment, multiple interconnect fabric modules ("IFMs") 110 with high-speed switching capabilities are used as intermediate routing devices to form an interconnect fabric, and multiple nodes 105, a network manager 115 and a Multi-Protocol Edge Switch ("MPEX") 120 are connected to the fabric. Each of the nodes has at least one VI NIC that uses virtual identifiers when communicating and receiving data. The MPEX is used to connect the Fibre Channel network to an external network, such as an Ethernet-based network, and similarly includes at least one VI NIC. Data is transmitted through the interconnect fabric using frames such as those defined by the Fibre Channel standard.

[0028] In this example embodiment, an IFM can be dynamically configured to interconnect its communications ports so that data can be transmitted through the interconnected ports. When the network manager receives a registration indication from a VI NIC for a data communication from a source node to a destination node, the network manager selects transmittal and response virtual identifiers to be used by the source and destination nodes when sending frames to each other. The network manager also identifies a path through the IFMs and their ports which frames will use when moving between the nodes. The network manager then configures the IFMs of the identified path so that when a frame that indicates the transmittal or response virtual identifiers is received at one of the IFMs, that frame is forwarded to the destination or source nodes via the path as appropriate. While the transmittal and response virtual identifiers thus use the same path (in opposite directions) in this example embodiment, they can use distinct paths in other embodiments.

[0029] Each IFM may maintain a virtual identifier table for each of its ports that maps virtual identifiers to its destinations ports. When a frame is received at a source port, the IFM then uses the virtual identifier for that frame and the virtual identifier table for the source port to identify a destination port through which the

frame is to be forwarded. Thus, in this embodiment, a virtual identifier identifies a path between devices, rather than identifying a source or a destination device. In one embodiment, a virtual identifier includes both a domain address and a virtual address. Each IFM is assigned a domain address, with the IFMs that are assigned the same domain address being in the same domain. The IFMs use the domain addresses to forward frames between domains, and the network manager may also configure the IFMs with inter-domain paths. When an IFM receives a frame whose virtual identifier has a domain address that matches its domain address, then the frame has arrived at its destination domain. The IFM then forwards the frame in accordance with the virtual address of the virtual identifier. If, however, the domain addresses do not match, then the frame has not arrived at its destination domain, and the IFM forwards the frame using an inter-domain path. The virtual identifier table for an IFM port may thus be divided in some embodiments into a domain address table and a virtual address table that respectively map domain addresses and virtual addresses to destination ports through which frames are to be forwarded.

[0030] As an illustrative example of using virtual identifiers for routing data communications, Figures 2A-2C illustrate an example of VI NIC embodiments using virtual identifiers to inter-communicate through an example Fibre Channel Fabric network. In particular, Figure 2A illustrates various VI NICs 250, 255, 270, 275 and 280 that are inter-communicating through a Fibre Channel fabric-based interconnect network that includes IFMs 262, 264 and 266. As discussed in greater detail below, Figure 2B illustrates a table containing information related to each of multiple example data communications discussed below.

[0031] A first example data communication begins when VI NIC 250, which is connected to port 0 of IFM 262, initiates a data communication to VI NIC 270, which is connected to port 25 of IFM 264. This data communication is indicated to be a persistent connection, and VI NIC 250 receives a transmittal virtual identifier A (*e.g.*, from a network manager for the network, not shown) to be used for routing communications to VI NIC 270. VI NIC 270 correspondingly receives a response

virtual identifier B to be used for routing responses to VI NIC 250. Those skilled in the art will appreciate that virtual identifiers can be represented in various formats, such as 24-bit identifiers in a Fibre Channel network. As previously discussed, VI NIC 250 is also notified of the response virtual identifier B supplied to VI NIC 270 so that VI NIC 250 can recognize communications received from VI NIC 270 as being part of the persistent connection and can forward those received data communications in an appropriate manner (*e.g.*, to an executing application (not shown) on the node to which VI NIC 250 belongs). VI NIC 270 similarly maps received data communications using virtual identifier A to an appropriate destination on the node to which VI NIC 270 belongs. The transmittal and response virtual identifiers A and B each correspond to a path through IFMs 262 and 264. In particular, data communications from VI NIC 250 using the transmittal virtual identifier A will be received at port 0 of IFM 262, and will be forwarded by that port along link 262a to output port 29 of IFM 262. That output port is statically connected to port 5 of IFM 264, which will receive the data communications using the transmittal virtual identifier A and will forward them along link 264a to output port 25 of IFM 264. VI NIC 270 will then receive the data communication. Data communications from VI NIC 270 to VI NIC 250 will return in a similar manner along that same path in an opposite direction.

[0032] As previously noted, Figure 2B illustrates a table reflecting the example data communications between the various VI NICs, and entries 1a and 1b of the table represent the dedicated connection between VI NICs 250 and 270 that was just discussed. As is shown, the determination of which of an associated pair of virtual identifiers is the transmittal virtual identifier and which is the response virtual identifier is made with respect to the source VI NIC using the virtual identifiers to route a data communication. Thus, VI NIC 250 uses virtual identifier A as its transmittal virtual identifier and virtual identifier B as its response virtual identifier for the dedicated connection, while VI NIC 270 uses them in an opposite matter.

[0033] After the connection between VI NICs 250 and 270 has ended, VI NIC 250 initiates a connectionless data communication to VI NIC 275, which is connected to port 28 of IFM 264. Transmittal and response virtual identifiers C and D are provided to VI NIC 250 to be used for the data communication, with the transmittal virtual identifier C corresponding to a path including link 262a from port 0 of IFM 262 to port 29 of IFM 262, followed by link 264b from port 5 of IFM 264 to port 28 of IFM 264. In the illustrated embodiment, a response virtual identifier is provided to VI NIC 250 so that response information can be provided to VI NIC 250 if necessary, such as an error message indicating that data communication was not successful. At least the transmittal virtual identifier C will also be provided to VI NIC 275 so that received data communications can be recognized and forwarded in an appropriate manner. Entry 2 of the table illustrated in Figure 2B corresponds to this data communication.

[0034] Immediately after the data communication to VI NIC 275, VI NIC 250 initiates a data communication to VI NIC 280, which is connected to port 20 of IFM 266. VI NIC 250 receives transmittal and response virtual identifiers E and F, with the path corresponding to transmittal identifier E including link 262c from port 0 to port 31 of IFM 262 and link 266a from port 0 to port 20 of IFM 266. Port 31 of IFM 262 is statically connected to port 0 of IFM 266. Entry 3 of the table illustrated in Figure 2B corresponds to this data communication.

[0035] Note that after this data communication, port 0 of IFM 262 is configured (barring any reconfigurations) to route data communications from VI NIC 250 that use any one of the transmittal virtual identifiers A, B or C. While port 0 forwards data communications in the illustrated embodiments for these transmittal virtual identifiers to different ports, that is not necessary. For example, port 0 could be configured to forward data communications for all the transmittal virtual identifiers to port 29 of IFM 262, and port 5 of IFM 264 could then be configured to forward data communications using transmittal virtual identifier C to port 10 of IFM 264 for communication to port 8 of IFM 266, which could then forward that received data communication to port 20 of IFM 266 for delivery to VI NIC 280.

[0036] VI NIC 275 next initiates a data communication that is determined (*e.g.*, by the network manager) should be routed to VI NICs 250 and 280. VI NIC 275 then receives a transmittal virtual identifier E and response virtual identifier G to be used for the data communication. When used by VI NIC 275, transmittal virtual identifier E corresponds to two paths through the network that lead to the two destination VI NICs. In particular, a path to VI NIC 250 includes link 264c from port 28 to port 6 of IFM 264 and link 262b from port 30 to port 0 of IFM 262. Port 6 of IFM 264 is statically connected to port 30 of IFM 262. The path from VI NIC 275 to destination VI NIC 280 includes links 264b from port 28 to port 10 of IFM 264 and link 266b from port 8 to port 20 of IFM 266. Port 10 of IFM 264 is statically connected to port 8 of IFM 266. Entry 4 of the table illustrated in Figure 2B corresponds to this data communication.

[0037] In this most recent data communication example, port 28 of IFM 264 is configured such that when it receives a data communication using the transmittal virtual identifier E from VI NIC 275, the port divides the received data communication and sends a copy of the data communication to both of ports 6 and 10 to IFM 264 for forwarding. Thus, this single transmittal virtual identifier is used to send a data communication to multiple destinations. Note, however, that it is not necessary that the port that initially receives the data communication (*i.e.*, port 28 of IFM 264 in this example) be the one to divide a received data communication into multiple copies. For example, port 28 of IFM 264 could instead be configured to send only a single copy of the received data communication to port 6 of IFM 264, and port 30 of IFM 262 could instead be configured to send a copy of the received data communication to both ports 0 and 31 of IFM 262. Alternately, port 28 of IFM 264 could be configured as initially discussed, but port 30 of IFM 262 could instead be configured to send copies of the received data communication to both ports 0 and 28 of IFM 262 if VI NIC 255 is determined to be another destination for the data communication.

[0038] Note also that the transmittal virtual identifier E used by VI NIC 275 in this most recent example data communication is identical to the transmittal virtual

identifier E previously used by VI NIC 250 for data communication to VI NIC 280. In this illustrated embodiment, the paths corresponding to virtual identifiers are relative to the source from which those data communications originate, and thus different VI NICs can use the same virtual identifier to correspond to different paths and to different destinations. This is possible since each of the ports of each of the IFMs can be separately configured in the illustrated embodiment to handle data communications having specified virtual identifiers. Thus, for example, port 28 of IFM 264 is configured to forward data communications received from VI NIC 275 that use the transmittal virtual identifier E to ports 6 and 10 of IFM 264, while port 0 of IFM 262 is configured to forward a data communication received from VI NIC 250 that uses the transmittal virtual identifier E to port 31 of IFM 262.

[0039] Figure 2C illustrates an example of a virtual identifier translation table used by VI NIC 250 when transmitting and receiving the example data communications. In the illustrated example, multiple applications programs are executing on a node to which VI NIC 250 corresponds and are using TCP/IP socket communication mechanisms to specify their data communications. In addition, in the illustrated example the VI NIC can identify various QOS communication parameters to be associated with each data communication. Each entry in the virtual identifier translation table corresponds to a distinct data communication of which the VI NIC has been notified. For example, entry 1 in the table corresponds to the previously discussed dedicated connection between VI NICs 250 and 270.

[0040] In this example, the data communication for entry 1 was initiated by an executing source application opening a TCP/IP socket having a destination of IP address "128.32.78.105" and a destination node software port of 3523, with this TCP/IP socket information stored in column 221 of the table. In addition, information is stored in column 223 of the virtual identifier translation table to enable received data communications to be forwarded to the appropriate executing application, which in this case is the source application. In this example, the VI NIC 250 determines (*e.g.*, based on the received indication of the

data communication) that the source application has source socket information that includes an IP address of "153.83.28.125" and a software port number of 3025 for the node on which the application is executing.

[0041] In this example, the VI NIC 250 also determines appropriate transmittal and response virtual identifiers for the data communication, as well as various QOS parameters related to the data communication (*e.g.*, by registering the data communication with a network manager that supplies the virtual identifiers and QOS parameters). The transmittal and response virtual identifiers are stored in columns 225 and 227 of the table respectively, and the QOS communications parameters are stored in one or more columns 229. In this example embodiment, the identified QOS communication parameters include a specified COS and an authorized minimum and maximum transmission priority. As shown, this data communication is assigned a COS of "1" (*e.g.*, which may correspond to dedicated connections) and a transmission priority range between 0 and 127 (*e.g.*, the full range for a 7-bit priority value).

[0042] In a similar manner, entries 2, 3 and 4 of the virtual identifier translation table correspond to example communications 2, 3 and 4 listed in the table illustrated in Figure 2B. As is shown, a single executing application may have multiple virtual identifier pairs shown in different entries of the virtual identifier translation table, such as entries 1 and 2 which share the same TCP/IP socket routing information in column 223. Conversely, even when multiple indicated data communications specify the same destination IP address, the data communications may be routed to different destination nodes, such as is shown with entries 1 and 3 of the table. More generally, in other embodiments the destination for an indicated data communication may be selected on the basis of information other than a specified TCP/IP destination socket, and if so the virtual identifier translation table would instead store in column 221 at least the minimal set of information needed to distinguish between the different data communications of which it is notified. For example, if indicated data communications have destinations selected based solely on a type of the executing application or on a type of the data being

transmitted, an indication of that type of information could be stored in column 221 instead of the destination TCP/IP socket information. In addition, in some embodiments an application could have multiple distinct virtual identifiers that can be used to communicate with a single destination, such as if the virtual identifiers are assigned to different paths through the network or have differing associated QOS parameters. Similarly, different applications on a source node could in various embodiments use the same or different virtual identifiers to communicate with a single destination, regardless of whether different virtual identifiers were assigned to different paths through the network.

[0043] Entry 4 in the virtual identifier translation table reflects a data communication initiated by a source other than VI NIC 250, in this case being VI NIC 275. In that situation, VI NIC 250 will store data in column 223 of the virtual identifier translation table indicating how to forward those received data communications (such as based on destination TCP/IP socket information included in a first of those received data communications), but need not store identification information for VI NIC 275 in column 221 since the example data communication is a 1-way connectionless transmittal. In addition, in the illustrated embodiment the transmittal virtual identifier E used by VI NIC 275 to route the data communication to VI NIC 250 is shown in column 227 of entry 4 of the virtual identifier translation table as being the response virtual identifier for the data communication, since from the perspective of VI NIC 250 the virtual identifier is used for received data communications. Those skilled in the art will appreciate that in other embodiments other types of information could be stored in the virtual identifier translation table (*e.g.*, connection preemption information) or existing types of information may not be present, and that the existing information could also be stored in other ways (*e.g.*, by having separate virtual identifier translation tables for outgoing and incoming data communications).

[0044] Figure 3 illustrates a node computing device 300 suitable for executing an embodiment of a VI NIC that uses virtual identifiers when transmitting and receiving data communications, as well as illustrating various other node

computing devices 350 with which node 300 can inter-communicate. The nodes are inter-connected through an Interconnect Fabric 380, and a Network Manager 370 is similarly connected to the Fabric.

[0045] The node computing device 300 includes a CPU 305, various I/O devices 310, storage 320 and memory 330. The I/O devices include at least one network interface 312 which connects the node to the Interconnect Fabric, as well as computer-readable media drive 313 and various other I/O devices 314. An embodiment of a VI NIC 340 is executing in memory, and it includes a Communication Registrar component 342, an Outgoing Communication Translator component 344 and an Incoming Communication Translator component 346. While the VI NIC in the illustrated embodiment includes multiple components executing in the main memory of the node, those skilled in the art will appreciate that other arrangements are possible in other embodiments, such as implementing a VI NIC together with a network interface on a single plug-in card that attaches to a bus for the node and that may include stand-alone memory and/or processing capabilities including hard-wired logic. In some embodiments, some or all of the VI NIC components may be a device driver for the node, such as for one of the network interfaces. In addition, those skilled in the art will appreciate that in other embodiments multiple VI NICs may be executing on a single node, such as to correspond to multiple network interfaces.

[0046] In the illustrated embodiment, multiple application programs 335 are also executing in memory, and can initiate or receive data communications with application programs executing on remote nodes. When one of the application programs initiates a data communication, the VI NIC is notified of the data communication, and the Communication Registrar component then registers that the communication with the Network Manager. In response, the Communication Registrar component receives a pair of transmittal and response virtual identifiers from the Network Manager as well as various QOS communication parameters for that data communication. In order to register a data communication with the Network Manager, the Communication Registrar component retrieves and uses

network manager communication parameters 327 from storage that may include a transmittal virtual identifier to route the communication to the Network Manager and a response virtual identifier to recognize the information received back from the Network Manager. The network manager communication parameters can be obtained in various ways, such as directly from the Network Manager during initialization of the node and/or the network. After the Communication Registrar component receives the virtual identifier pair and other information for the registered data communication, it stores that information in the virtual identifier translation table 325 on storage for use when transmitting and receiving data communications.

[0047] When an application program is ready to perform a data communication, the Outgoing Communication Translator receives notification of the communication to be performed. If the initial notification used by the Communication Registrar to initiate registration was itself an indication to perform a communication, the Outgoing Communication Translator component can receive this notification from the Communication Registrar component after the registration has been completed. The Outgoing Communication Translator component analyzes the information provided about the data communication to be performed, maps that data communication to a corresponding entry in the virtual identifier translation table in order to determine the appropriate transmission information to be used for the data communication, and then transmits the data using the information retrieved from the virtual identifier translation table. Those skilled in the art will appreciate that the Outgoing Communication Translator may also need to perform additional formatting of the data to be transmitted, such as to generate one or more appropriate Fibre Channel frames for the illustrated example in which the network is a Fibre Channel Interconnect Fabric. In addition, in some embodiments the Outgoing Communication Translator component may verify the accuracy of the communication indicated by the application program before transmitting the communication, such as to ensure that a priority requested

by the application program to be used for the transmission falls within the transmission priority bounds assigned to the data communication.

[0048] In a similar manner, the Incoming Communication Translator component is notified when the network interface receives incoming data communications that are routed using virtual identifiers. Upon receiving notification of such a received data communication, the Incoming Communication Translator determines the transmittal virtual identifier used to route the data communication to the node and uses the virtual identifier translation table to map that virtual identifier to one or more of the application programs executing in memory. Upon determining one or more appropriate application programs to receive the data communication, the VI NIC then forwards the received data communication to those application programs.

[0049] In the case of a received data communication that is a response to a data communication initiated at node 300, the necessary information for forwarding the received data communication will already be present in the virtual identifier translation table based on the Communication Registrar component having previously registered that data communication. In the case of data communications that are initiated at another node, however, the necessary information to forward the received data communication to an executing application program may or may not already be present in the virtual identifier translation table. For example, in some embodiments the Network Manager will have supplied the appropriate information to the VI NIC (*e.g.*, to the Communication Registrar component) before the data communication is received, and the information could be stored in the virtual identifier translation table at that time. In other embodiments, the appropriate information for forwarding the received data communication may be added to the virtual identifier translation table at the time that the data communication is received, such as by the Incoming Communication Translator component analyzing information included in the data communication to identify the needed information.

[0050] In some embodiments, the Incoming Communication Translator component will also process received data communications in various ways before forwarding them to one or more appropriate application programs. For example, in some embodiments application programs may expect received data communications to include information specific to the receiving application, such as one or more network addresses associated with that application. If the VI NIC has access to the appropriate information for the application, such as from the virtual identifier translation table, the Incoming Communication Translator component can add that information to a received data communication when it is missing or incorrect (or for every received data communication). For example, when the executing applications are using TCP/IP socket mechanisms or more generally receiving data in the form of IP packets, the Incoming Communication Translator component could ensure that the data communication forwarded to an executing application includes the appropriate IP address and/or port number associated with that application. In addition, those skilled in the art will appreciate that the Incoming Communication Translator component may need to reformat received information into an appropriate form for the application receiving the information, such as by converting a received Fibre Channel frame into one or more IP packets.

[0051] Those skilled in the art will also appreciate that node computing device 300 is merely illustrative and is not intended to limit the scope of the present invention. Computing device 300 may be connected to other devices that are not illustrated, including one or more networks such as the Internet or via the World Wide Web. In addition, computing device 300 could be one part of an EDN, such as by being a device at any one or more of the EDN sub-networks. Various available products could be used as network interfaces and/or to implement some or all of the functionality of a VI NIC, including products from Banderacom, Inc. and Mellanox Technologies. Those skilled in the art will also appreciate that the functionality provided by the illustrated VI NIC components may in some embodiments be combined in fewer components or distributed in additional components. Similarly,

in some embodiments, the functionality of some of the illustrated components may not be provided and/or other additional functionality may be available.

[0052] Those skilled in the art will also appreciate that, while various items are illustrated as being stored in memory while being used, these items or portions of them can be transferred between memory and other storage devices for purposes of memory management and data integrity. Similarly, items illustrated as being present on storage while being used can instead be present in memory and transferred between storage and memory. Some or all of the components and data structures may also be stored (*e.g.*, as instructions or structured data) on a computer-readable medium, such as a hard disk, a memory, a network, or a portable article to be read by an appropriate drive. The components and data structures can also be transmitted as generated data signals (*e.g.*, as part of a carrier wave) on a variety of computer-readable transmission mediums, including wireless-based and wired/cable-based mediums. Accordingly, the present invention may be practiced with other computer system configurations.

[0053] Figure 4 is a flow diagram of an embodiment of the Communication Registrar routine 400. The routine receives indications of new data communications from either a local executing application or from a remote network manager, registers new data communications indicated by local applications with the network manager and receives appropriate virtual identifiers and other information to be used for the data communication in response, and stores any received information from the network manager in the virtual identifier translation table for use in processing incoming and outgoing data communications.

[0054] The routine begins at step 405 where an indication is received of a new communication. The routine continues to step 410 where it determines if the indication was received from the network manager, such as for a data communication initiated by a remote source. If the data communication is instead from a local executing application, the routine continues to step 415 to determine whatever information about the data communication that will be used to register

the data communication with the network manager so that the network manager can determine appropriate destinations for the data communication. In the illustrated embodiment, information is determined about the type of data to be communicated, about the application that initiated the communication, and about any destination information specified by the local application.

[0055] The routine then continues to step 420 where it is determined if the indicated data communication needs to be registered with the network manager. For example, if the indicated data communication is a transmittal of data for an existing persistent connection, the connection will already have been registered and registration is not necessary. In other situations, even a new data communication may not need to be registered, such as a data communication that will be communicated in the same manner and to the same destination as a previous communication, as the information provided for the previous data communication may be able to be reused. In particular, the routine compares whenever information is used to uniquely identify the destination and/or the manner of transmission for the indicated new data communication, and determines if there is a match for that information already present in the virtual identifier translation table. Those skilled in the art will appreciate that in other embodiments the routine may not make this determination and instead send registration information to the network manager for each new indicated data communication, such as if the network manager provides functionality to determine whether to reuse previously provided transmission information or to instead send new transmission information.

[0056] If it is determined in step 425 that the new indicated data communication needs to be registered, the routine continues to step 430 to send a communication registration notification to the network manager that includes the determined communication type information. The routine then continues to step 435 to receive from the network manager an indication of a pair of transmittal and response virtual identifiers for the data communication, as well as optionally receiving other communication parameters to be used as part of the

communication. The routine then continues to step 440 to store the received information in the virtual identifier translation table, as well as to optionally store routing information with which to route a response data communication back to the application that initiated this new data communication. After step 440, or if it was instead determined in step 425 that the new indicated data communication did not need to be registered, the routine continues to step 490 to determine if there are more indications to receive. If so, the routine returns to step 405, and if not the routine continues to step 499 and ends.

[0057] If it was instead determined in step 410 that the received indication of the new data communication is from the network manager, the routine continues to step 445 to receive transmission information related to that new data communication. In particular, in the illustrated embodiment the network manager will supply information about a pair of transmittal and response virtual identifiers to be used to route the indicated data communication to the node and to be used to route any responses back to the originating node, and will optionally also supply other communication parameters that will be used as part of the data communications. The routine then continues to step 450 to determine a local application to which the incoming communication is to be forwarded, such as based on information supplied by the network manager (*e.g.*, TCP/IP socket information provided by the source application initiating the new data communication). After step 450, the routine continues to step 440 to store the received information from the network manager and the routing information for the local application in the virtual identifier translation table.

[0058] Figure 5 is a flow diagram of an embodiment of the Outgoing Communication Translator routine 500. The routine receives indications of outgoing data communication, determines an appropriate transmittal virtual identifier to be used for routing the data communication as well as optionally determining other communication parameters to be used, and transmits the data communication using the determined transmittal virtual identifier and other determined communication parameters.

[0059] The routine begins at step 505 where an indication is received of an outgoing communication. The routine continues to step 510 to execute a subroutine to verify that the communication transmittal is authorized. In other embodiments, this verification step may not be performed. The routine next continues to step 515 to determine if the communication transmittal was determined to be authorized, and if not continues to step 520 to send an error message to the application that initiated the data communication. If it was instead determined in step 515 that the communication transmittal was authorized or if no authorization verification was performed, the routine continues to step 525 to map communication type information for the indicated data communication to a corresponding entry in the virtual identifier translation table for a registered data communication. For example, in some embodiments the communication type information to be used for the mapping may be destination TCP/IP socket information provided by the application that initiated the data communication.

[0060] The routine then continues to step 530 to format the outgoing data communication in a manner appropriate for the network type being used and to use the transmittal virtual identifier and any other communication parameters indicated in the virtual identifier translation table entry. In the illustrated embodiment, one or more Fibre Channel frames are generated by storing response and transmittal virtual identifiers in the Fibre Channel frame locations for source and destination identifiers and by storing the data to be communicated as the payload of the frames, and the header information for the frames is specified to correspond to a Fibre Channel COS and a priority that is consistent with the information in the virtual identifier translation table entry. The routine then continues to step 535 to send the generated frames to the local IFM attached to the hardware output port to which the VI NIC corresponds. After step 535 or step 520, the routine continues to step 590 to determine if there are more indications to be received. If so, the routine returns to step 505, and if not the routine continues to step 599 and ends.

[0061] Figure 6 is a flow diagram of an embodiment of the Verify Communication Transmittal subroutine 600. The subroutine receives an indication of a data communication and determines if the data communication is consistent with corresponding information that was stored in the virtual identifier translation table based on a prior registration for the data communication. The subroutine begins at step 605 where an indication is received of a communication to be transmitted. The subroutine continues to step 610 to determine if the virtual identifier translation table has an entry corresponding to the communication. In step 615, if there was not a corresponding entry, the subroutine to step 635 to return an indication that the data communication is not authorized. If it is instead determined in step 615 that a corresponding entry was present, the subroutine continues to step 620 to determine if the manner of the transmittal of the data communication is consistent with the transmission information in the corresponding entry, including use of virtual identifiers and other communication parameters. In some embodiments, it is verified that the data communication includes a pair of transmittal and response virtual identifiers that were provided together by the network manager and that the COS and priority for the data communication corresponds to the specified COS and priority limits provided by the network manager. If in step 625 it is determined that the data communication transmittal was not consistent, the subroutine continues to step 635. If the data communication is instead determined to be consistent, the subroutine continues to step 630 to return an indication that the data communication is authorized. After step 630 or 635, the subroutine continues to step 699 and ends.

[0062] Figure 7 is a flow diagram of an embodiment of the Incoming Communication Translator routine 700. The routine receives indications of incoming data communications and forwards those data communication to appropriate local destinations such as an executing application. In some situations, the routine can also modify the incoming data communication in various ways, such as to replace missing or incorrect destination application-specific information with appropriate information.

[0063] The routine begins at step 705 where an indication is received of an incoming data communication. The routine continues to step 710 to execute a subroutine to determine whether the data communication is authorized. Those skilled in the art will appreciate that in some embodiments such data communication verification may not be performed. In step 715, the routine then determines if the data communication is authorized, and if not continues to step 720 to send an error message to the sender of the communication. If it is instead determined in step 715 that the data communication is authorized or if the verification is not performed, the routine continues to step 725 to map the transmittal virtual identifier (or in some embodiments the pair of the transmittal and response virtual identifiers) used to route the data communication to a corresponding entry in the virtual identifier translation table.

[0064] In step 730, the routine then determines if the data communication includes information specific to the destination application that is incorrect or is missing, such as a network address. If so and if the correct information is accessible, such as by being stored in the virtual identifier translation table, the routine then continues to step 735 to replace the included incorrect information with the correct information. Those skilled in the art will appreciate that in other embodiments, if correct destination application-specific information is accessible it could always be used to replace information sent in a received data communication without checking if the included information is missing or incorrect. After step 735, or if it was instead determined that in step 730 that incorrect information was not included or that correct information to be used as a replacement was not accessible, the routine continues to step 740 to forward the received data communication to the appropriate local destination by using the routing information from the corresponding entry in the virtual identifier translation table. In addition, the routine may format the data communication in a manner appropriate for the local destination, such as by converting the received data communication into IP packet format. The routine then continues to step 795 to

determine if there are more indications to be received. If so, the routine returns to step 705, and if not the routine continues to step 799 and ends.

[0065] Those skilled in the art will also appreciate that in some embodiments the functionality provided by the routines discussed above may be provided in alternate ways, such as being split among more routines or consolidated into less routines. Similarly, in some embodiments illustrated routines may provide more or less functionality than is described, such as when other illustrated routines instead lack or include such functionality respectively, or when the amount of functionality that is provided is altered. Those skilled in the art will also appreciate that the data structures discussed above may be structured in different manners, such as by having a single data structure split into multiple data structures or by having multiple data structures consolidated into a single data structure. Similarly, in some embodiments illustrated data structures may store more or less information than is described, such as when other illustrated data structures instead lack or include such information respectively, or when the amount or types of information that is stored is altered.

[0066] From the foregoing it will be appreciated that, although specific embodiments have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims. In addition, while certain aspects of the invention are presented below in certain claim forms, the inventors contemplate the various aspects of the invention in any available claim form. For example, while only one some aspects of the invention may currently be recited as being embodied in a computer-readable medium, other aspects may likewise be so embodied. Accordingly, the inventors reserve the right to add additional claims after filing the application to pursue such additional claim forms for other aspects of the invention.